motion into the units such as arrow key clicks or mouse clicks expected by existing host computer systems.

[0265] Step **675** of chord motion conversion simply picks the first slice in the given chord activity structure for processing. Step **676** scales the current values of the extracted velocity components by the slice's motion sensitivity and acceleration parameters. Step **677** geometrically projects or clips the scaled velocity components into the slice's defined speed and direction range. For the example mouse cursor slice, this might only involve clipping the rotation and scaling components to zero. But for an arrow key slice, the translation velocity vector is projected onto the unit vector pointing in the same direction as the arrow. Step **678** integrates each scaled and projected component velocity over time in the slice's accumulators until decision diamond **680** determines at least one unit of motion has been accumulated. Step **682** looks up the slice's preferred mouse, key, or three-dimensional input event format, attaches the number of accumulated motion units to the event; and step **684** dispatches the event to the outgoing queue of the host communication interface **20**. Step **686** subtracts the sent motion events from the accumulators, and step **688** optionally clears the accumulators of other slices. If the slice is intended to generate a single key command per hand motion, decision diamond **689** will determine that it is a one-shot slice so that step **690** can disable further event generation from it until a slice with a different direction intervenes. If the given slice is the last slice, decision diamond **692** returns to step **650** to await the next scan of the sensor array. Otherwise step **694** continues to integrate and convert the current motion for other slices.

[0266] Returning to FIG. **40**A, for some applications it may be desirable to change the selected chord whenever an additional finger touches down or one of the fingers in the chord lifts off. However, in the preferred embodiment, the selected chord cannot be changed after slide initiation by asynchronous finger touch activity. This gives the user freedom to rest or lift addition fingers as may be necessary to get the best precision in a desired degree of freedom. For example, even though the finger pair chord does not include the thumb, the thumb can be set down shortly after slide initiation to access the full dynamic range of the rotation and scaling degrees of freedom. In fact, all remaining lifted fingers can always be set down after initiation of any chord to allow manipulation by the whole hand. Likewise, all fingers but one can be lifted, yet translation will continue.

[0267] Though asynchronous finger touch activity is ignored, synchronized lifting and pressing of multiple fingers subsequent to slide initiation can create a new synchronized subset and change the selected chord. Preferably this is only allowed while the hand has paused but its fingers are still resting on the surface. Decision diamond **670** will detect the new subset and commence motion testing in decision diamond **673** which is analogous to decision diamond **656**. If significant motion is found in all fingers of the newly synchronized subset, step **674** will select the new subset as the slide chord and lookup a new chord activity structure in analogy to step **658**. Thus finger synchronization again allows the user to switch to a different activity without forcing the user to lift the whole hand from the surface. Integration of velocity components resumes but the events generated from the new chord activity structure will presumably be different.

[0268] It is advantageous to provide visual or auditory feedback to the user about which chord activity structure has been selected. This can be accomplished visually by placing a row of five light emitting diodes across the top of the multi-touch surface, with one row per hand to be used on the surface. When entering slide mode, step **658** would turn on a combination of these lights corresponding to the combination of fingers in the selected chord. Step **674** would change the combination of active lights to match the new chord activity structure should the user select a new, chord, and step **668** would turn them off. Similar lights could be emulated on the host computer display **24**. The lights could also be flashed to indicate the finger combination detected during chord taps in step **636**. The implementation for auditory feedback would be similar, except light combinations would be replaced with tone or tone burst combinations.

[0269] The accumulation and event generation process repeats for all array scan cycles until decision diamond **664** detects liftoff by all the fingers from the initiating combination. Decision diamond **666** then checks the pre-liftoff deceleration flag of the dominant motion, component. The state of this flag is determined by step **556** or **558** of translation extraction (FIG. **37**) if translation is dominant, or by corresponding flags in step **534** of polar extraction. If there has been significant deceleration, step **668** simply exits the chord slide mode, setting the selected chord to null. If the flag indicates no significant finger deceleration prior to liftoff, decision diamond **666** enables motion continuation mode for the selected chord. While in this mode, step **667** applies the pre-liftoff weighted average (**560**) of dominant component velocity to the motion accumulators (**678**) in place of the current velocities, which are presumably zero since no fingers touch the surface. Motion continuation mode does not stop until any of the remaining fingers not in the synchronized subset are lifted or more fingers newly touch down. This causes decision diamond **664** to become false and normal slide activity with the currently selected chord to resume. Though the cursor or scrolling velocity does not decay during motion continuation mode, the host computer can send a signal instructing motion continuation mode to be canceled if the cursor reaches the edge of the screen or end of a document. Similarly, if any fingers remain on the surface during motion continuation, their translations can adjust the cursor or scrolling velocity.

[0270] In the preferred embodiment, the chord motion recognizers for each hand function independently and the input events for each chord can be configured independently. This allows the system to allocate tasks between hands in many different ways and to support a variety of bimanual manipulations. For example, mouse cursor motion can be allocated to the fingertip pair chord on both hands and mouse button drag to a triple fingertip chord on both hands. This way the mouse pointer can be moved and drug with either hand on either half of the surface. Primary mouse clicks would be generated by a tap of a fingertip pair on either half of the surface, and double-clicks could be ergonomically generated by a single tap of three fingertips on the surface. Window scrolling could be allocated to slides of four fingers on either hand.

[0271] Alternatively, mouse cursor manipulations could be allocated as discussed above to the right hand and right half of the surface, while corresponding text cursor manipu-